Optimization Transfer and Projective Scaling Strategy for $l_p$ Loss with Convex Penalizations

Hongbo Zhou  
Department of Computer Science  
Southern Illinois University Carbondale  
IL 62901, USA  

Qiang Cheng  
Department of Computer Science  
Southern Illinois University Carbondale  
IL 62901, USA

Abstract

This paper aims at constructing an effective, generic, and generally applicable procedure for minimizing $l_p$ loss functions with convex penalizations, which represent many statistical models and reoccur in numerous machine learning problems. In this paper, the effectiveness means high accuracy, efficiency, and scalability; genericness requires few modifications for closely related penalizations, such as any functions from the $l_p$ family; and applicability demands few restrictions on problem formulations such as under-determination and over-determination. Previous work considers some of these properties but does not realize all of them simultaneously. Based on $l_p$ functions’ structure and a double-layer optimization transfer technique, the proposed method, a generic iterative procedure (GIP), is proved to be globally convergent for convex formulations and admits a linear convergence rate. Besides convex penalizations, GIP can also deal with some non-convex penalization functions such as $l_0$. Additionally, as a featured application, we introduce an empirical yet effective reformulation for penalized low-rank matrix approximation problems with missing data, and this reformulation enables us to use GIP to investigate many matrix decomposition problems as well. The effectiveness, genericness, and applicability of the proposed method are verified through extensive experiments on sparse signal recovery, image-based object recognition, sparse dictionary learning-based image denoising, and real-world structure-from-motion problems. Our work expands previously known algorithmic results on penalized $l_p$ learning problems.

Keywords: Convex function, generic iterative procedure, $l_p$ loss function, low-rank matrix approximation, matrix decomposition, missing data, penalization, regularization, structure from motion

1. Introduction

1.1 $l_p$ Learning Problems Are Important

The penalized $l_p$ loss functions (or simply $l_p$ learning problems) have reoccurred frequently in recent advances in face classification (Wright et al., 2009), dictionary learning (Aharon et al., 2006), low-rank matrix decomposition (Eriksson and Hengel, 2010), etc. Given a data
matrix $A \in \mathcal{R}^{m \times n}$ and a vector $y \in \mathcal{R}^m$, we consider the following penalized $l_p$ learning problems:

$$\hat{w} = \arg\min_{w \in \mathcal{D}} \{L(w) + \lambda f(w)\}, \quad (1)$$

where $\mathcal{D} \subset \mathcal{R}^n$ is the domain, $L(w) = \frac{1}{p}||y - Aw||_p^p$ is the $l_p$ loss function, $p \in [1, 2]$ and $||\cdot||_p$ is a vector norm, $\lambda \in \mathcal{R}^+$ is a weight, and $f(w) : \mathcal{D} \rightarrow [0, +\infty)$ is a nonnegative penalization function. Unless otherwise specified, we require $\mathcal{D}$ be a bounded closed set and $f(w)$ be a coercive convex function. Equation (1) represents many penalized regression or learning problems, such as ridge regression or Tikhonov regularization (Hoerl and Kennard, 1988; Frank and Friedman, 1993), lasso (Tibshirani, 1996), elastic net (Zou and Hastie, 2005), local coordinate coding (Yu et al., 2009), etc. Especially, the case of $p = 1$ was shown to achieve state-of-the-art performance in structure-from-motion (SFM) problems (Eriksson and Hengel, 2010).

1.2 Why a New Algorithm

Despite extensive research effort, effective, generic, and generally applicable algorithms for solving Equation (1) are still needed, especially for non-differentiable cases. In this paper, the effectiveness means high accuracy, efficiency, and scalability; genericness requires few modifications for closely related penalizations, such as any functions from the $l_p$ family; and applicability demands few restrictions on problem formulations such as under-determination and over-determination. Previous work considers some of these properties but does not realize all of them simultaneously. Various well-known algorithms have different trade-offs among these properties. Taking the most extensively studied lasso formulation ($p = 2$ and $f(w) = ||w||_1$ in Equation (1)) for example, standard convex programming techniques such as interior-point often have the best accuracy and applicability but they usually lack efficiency and scalability; Fixed-point Continuation and Active Set (FPC-AS) method (Wen et al., 2009) is accurate and efficient, but it is only applicable to under-determined systems; Gradient Projection for Sparse Reconstruction (GPSR) method (Figueiredo et al., 2007) is efficient, scalable and generally applicable, but its accuracy is inferior to FPC-AS and convex programming techniques. We will show a challenging application of low-rank matrix approximation with missing data which has three prominent features: 1) It requires high scalability and thus conventional convex programming techniques are inappropriate; 2) It demands high reconstruction accuracy and thus GPSR is not suitable; 3) The resulted formulation is often over-determined and thus FPC-AS is inapplicable. Such challenging problems demand further study of these well-known formulations. A more important reason for this work is that for some other important convex models, such as the $l_1$ loss function in Eriksson and Hengel (2010), effective algorithms are still unavailable. Moreover, aside from these convex models, for non-convex models, such as the $l_0$ penalization, few algorithms are known to work. Considering these issues and challenges, we aim at developing an effective, generic, and generally applicable algorithm for solving Equation (1), and we also show how to relax it to deal with some non-convex models such as the $l_0$ penalization.
1.3 Proposed Approach, Main Steps

For Equation (1), well-known efficient and scalable algorithms mainly consider various approximation schemes, such as gradient projection (Figueiredo et al., 2007), continuation (Wen et al., 2009) or smoothing (Becker et al., 2009) techniques. These approximation schemes are often derived solely based on the penalization terms. To our knowledge, there is still little work on how to construct consistent and effective approximation schemes for the $l_p$ loss function and the penalization term simultaneously. Consistency is critical here because it avoids possible alternative approximations for loss and penalization terms. We show in this paper that there exists such a scheme for $l_p$ loss function with convex penalizations. By employing consistent approximations, we expect a better algorithm with potentially higher accuracy than the previous ones based on alternative approximations, while maintaining a similar level of efficiency and scalability. The proposed approach mainly consists of two parts, building a consistent approximation scheme for the $l_p$ loss function and the penalization term, and constructing efficient practical algorithms by introducing effective approximation schema.

In the first part, we introduce a double-layer optimization transfer technique to build theoretical foundations for the proposed generic algorithms. The first layer is to construct surrogate functions for the $l_p$ loss and the coercive convex penalization function $f(w)$ respectively. The surrogate functions are shown to be uniformly convergent to original functions. With a careful design of surrogate functions, we show that the second layer of optimization transfer, which is based on the well-known majorization-minimization technique (Lange et al., 2000), can be adopted to construct a Generic Iterative Procedure (GIP) for solving Equation (1). We prove the global convergence of the proposed GIP algorithm and show that it admits a linear convergence rate. This part aims at building theoretically exact GIP for all convex penalization functions. It involves an adjusted Hessian matrix of the penalization term, which might be difficult to evaluate in practice.

In the second part, we construct effective approximation schemes to build efficient and scalable practical algorithms for some component-wise separable penalization functions, such as the $l_q$ family with $q \in [0, 2]$. For these functions, a form of the Karmarkar’s projective scaling strategy (Pino, 1989; Stone and Tovey, 1991) can be constructed so that the adjusted Hessian matrix can be evaluated only using the first-order gradient information. This projective scaling strategy leads to an important connection which associates each iteration of the GIP with a perturbed Tikhonov regression problem. This association has at least two important implications: developing scalable algorithms and empirically understanding sparsity promoting property of the regularization terms. First, we can leverage efficient ridge regression algorithm (Turlach, 2006) or approximate least squares algorithms to solve the original Equation (1). By doing so, we may enhance the scalability of the GIP readily when scalable least squares algorithms are used. Second, we show that simple analysis of the perturbed problems can contribute to empirical understanding of some important properties of the regularization term, such as why $l_q$ regularizations with $q \in [0, 2)$ can promote sparsity. We note that Karmarkar’s projective scaling strategy can also be constructed from some non-convex penalization functions, such as $l_q$ with $q \in [0, 1)$. For non-convex situations, there is no global convergence guarantee for the GIP and we show that it always converges to a stationary point. More discussions are given in Section 4. The approximation
is mainly due to two aspects: inaccurate solving of Tikhonov regression problems and heuristic strategies for choosing initialization points for non-convex penalizations.

1.4 Typical Application: Missing Data Low-rank Matrix Approximation

With aforementioned approach, we proceed to consider an important and challenging application and compare the GIP with a number of state-of-the-art methods. In fact, penalized low-rank matrix approximation problems, especially those involving missing data, are of great interest in many important applications such as structure-from-motion (SFM) (Tomasi and Kanade, 1993; Eriksson and Hengel, 2010), recommendation systems (Takács et al., 2009), and dictionary learning (Aharon et al., 2006). In this part, we adopt a matrix decomposition-based formulation to solve the SFM problem. We compare the GIP with those matrix-decomposition-based state-of-the-art methods. Especially, Wiberg (Wiberg, 1976) is a baseline method, and $l_1$ Wiberg (Eriksson and Hengel, 2010) represents a recent award-winning advance. We also compare with other existing work, which is mainly focused on some simple regularizations such as a $F$ norm (Buchanan and Fitzgibbon, 2005), or equivalently, trace norm. The implication of our work on this application is two-fold. First, we develop an effective framework for approximately solving penalized missing data low-rank matrix approximation problems; second, due to its requirements for high accuracy and scalability, this low-rank approximation problem also serves as a good example to differentiate the GIP from other state-of-the-art algorithms, such as GPSR and $l_1$ Wiberg.

1.5 Experimental Results

At last, we verify the effectiveness, genericness, and applicability of the proposed GIP through simulations (such as sparse signal recovery and low-rank matrix approximation) and extensive experiments on image-based face recognition, sparse dictionary-learning based image denoising, and real-world structure-from-motion problems. We compare the GIP with several state-of-the-art methods including a standard convex programming-based method and various approximation-based methods (with either $l_1$ or $l_0$ regularization terms), and we conclude that: 1) the GIP is accurate, efficient, and scalable on the aforementioned real-world learning problems; 2) the GIP is generic and can be applied to many penalization functions subject to a change of the Hessian matrix; 3) the GIP is generally applicable to either over-determined or under-determined systems; 4) GIP is also applicable to some non-convex regularization functions such as $l_0$. Especially, on the challenging real-world penalized missing data low-rank approximation problem, the GIP outperforms the state-of-the-art algorithms in both accuracy and scalability.

1.6 Summary of Contributions and Organizations

As a brief summary, our contributions include, but are not limited to, the following:

- We propose an effective, generic, and generally applicable iterative procedure for solving Equation (1). For coercive convex penalization functions, we prove the global convergence of the GIP and characterize its convergence rate. Different from existing
Optimization Transfer for $l_p$ Loss with Convex Penalizations

- We reveal the connection between GIP and perturbed Tikhonov regression problems. This connection not only enables us to construct highly efficient and scalable GIP solvers, but contributes to the understanding of important properties of various penalization procedures.

- We show that the GIP is also applicable to many non-convex or even discontinuous penalization terms such as $l_0$. Although for general non-convex situations there is no guarantee for its global convergence, we prove that the GIP always converges to a stationary point. Its applicability in these situations is also verified by extensive experiments.

- As important applications, we incorporate the GIP into a missing data low-rank matrix approximation framework and this extends algorithmic investigations on many challenging matrix-form problems. Previously, the studies on penalized low-rank matrix approximation were mainly focused on trace-norm based (or equivalently, $F$ norm based) penalizations. In contrast, our methods allows for more general regularization terms. We show by real-world structure-from-motion experiments that simple $l_1$ or $l_0$ regularization terms work well in combating strong outliers.

The reminder of this paper is organized as follows. We start by reviewing some closely related work in Section 2 and proceed to introduce the GIP in Section 3. The first two subsections of Section 3 discuss how to construct a first-layer optimization transfer for the $l_p$ loss and the regularization term $f(w)$ respectively; and the rest of this section is devoted to the second-layer optimization transfer, proof of convergence, discussions on the convergence rate. In Section 4, we introduce a form of the Karmarkar’s projective scaling strategy, establish the connection between the GIP and the perturbed Tikhonov regression problems, and provide some convergence results for non-convex situations. In Section 5, we give an effective reformulation for penalized missing data low-rank matrix approximation problems. In Section 6, we design and carry out numerical simulations and comprehensive experiments on sparse signal recovery, image-based object recognition, low-rank matrix approximation recovery, and real-world structure-from-motion problems. Finally, we conclude this paper and discuss some future research topics in Section 7.

2. Related Work

We briefly review some closely related work in several aspects: the formulation of Equation (1), some well-known special cases of Equation (1), iterative re-weighted least squares algorithms, smoothing surrogates for convex functions, optimization transfer and the majorization-minimization framework.

This work is focused on optimization aspect of Equation (1) and it does not assume the over-determination or under-determination of the system. This makes the proposed method generally applicable. Often times, the penalization terms are imposed to reduce the sensitivity in the least-squares problem and generate certain desired properties (Tibshirani, 1996; Donoho and Tsaig, 2008). For example, Tibshirani (1996), and Osborne
et al. (2000a,b) are on the penalization of over-determined systems; Donoho and Tsag (2008) is on the penalization of under-determined systems. Also, this work only deals with unconstrained optimization formulation. Although many applications can be solved with corresponding constrained optimization formulations, Equation (1) still has some advantages especially when an explicit penalization weight can be constructed for avoiding cross-validations (Dhillon et al., 2010).

Probably the most well-known cases of Equation (1) are ridge regression (Hoerl and Kennard, 1988; Frank and Friedman, 1993) and lasso (Tibshirani, 1996). In the image processing area, \( l_2/l_1 \) Total Variation (TV) regularizations play an important role in many real-world applications (Chartrand and Yin, 2008). Also, many learning problems involve matrix forms of Equation (1), such as dictionary learning (Aharon et al., 2006), structure-from-motion (Tomasi and Kanade, 1993), etc. Starting from the work of Tomasi and Kanade (1993), matrix factorization based SFM has attracted much attention (Buchanan et al., 2005; Eriksson and Hengel, 2010; Mitra et al., 2010). Two issues make this problem challenging. First, factorization-based SFM requires high accuracy in recovering camera and structure matrix; second, real-world SFM applications often involve a large number of missing components and heavy outliers. Especially, the presence of missing components deteriorates the orthographic property of SFM given in Tomasi and Kanade (1993) because it is impossible to correctly register the measurement matrix. Moreover, the unavoidable outliers from feature points or tracks further compound the high accuracy requirement. We will show that the recent advance such as Eriksson and Hengel (2010) is still inefficient and not scalable.

Although many techniques are known for tackling the \( l_p \) loss function and the convex penalization function individually, to our knowledge, there is no known result on the general form of Equation (1). Historically, the \( l_p \) loss function (without the penalization term) occurs frequently in robust linear regression related studies (Huber, 1981; Scales et al., 1988; Scales and Gersztenkorn, 1988). Most of the well-known techniques are essentially based on either iterative re-weighted least squares (IRLS) (Weisfeld, 1937; Schlossmacher, 1973; Holland and Welsch, 1977; Scales and Gersztenkorn, 1988) or smooth surrogate function based techniques (Vogel and Oman, 1996; Bissantz et al., 2009). On the other side, many techniques have been developed for minimizing a general convex function \( f(w) \). Conventionally, various approaches are based on techniques (or combinations of them) such as the continuation, subgradient method, subgradient-projection, bundle method (Shor, 1985; Bertsekas, 1999), and proximal splitting methods (Combettes and Pesquet, 2011); see (Nurminski, 1982; Boyd and Vandenberghe, 2004; Combettes and Pesquet, 2011) for detailed comments. As aforementioned, many well-known algorithms are generated by applying these techniques to \( l_1 \) penalized least squares problems. For example, GPSR (Figueiredo et al., 2007) uses gradient-projection approximation, FPC-AS (Wen et al., 2009) relies on some continuation schemes, NESTA (Becker et al., 2009) is built upon smooth techniques, iterative thresholding (Blumensath and Davies, 2008) is an example of proximal splitting method, etc. It shall be noticed that many algorithms require additional constraints on problem formulations. For example, FPC-AS only works for under-determined systems, and iterative thresholding requires \( ||A||_F \leq 1 \). Due to the iterative thresholding’s unsatisfying convergence property in
practice and inconsistency in penalization weight\(^2\), we mainly compare with GPSR, FPC-AS, NESTA-UP (an unconstrained programming version of NESTA) for lasso formulations. Aside from these convex formulations, the \(l_0\) penalization has shown some advantages (Lin et al., 2008) over the \(l_1\) penalization but there are only few algorithms available for approximating the \(l_0\). Besides the smoothing-based SL0 (Mohimani et al., 2009), other well-known methods include ISD (Wang and Yin, 2010) and re-weighted \(l_1\) (Candés et al., 2008). We note that ISD is only applicable to under-determined systems, and the re-weighted \(l_1\) is computationally expensive because it requires solving a lasso problem in each iteration.

The iterative re-weighted least squares (IRLS) algorithm was originally developed for robust linear regression problems (Weiszfeld, 1937; Schlossmacher, 1973; Holland and Welsch, 1977; Scales and Gersztenkorn, 1988) without any penalization terms, or a constrained programming version (Gorodnitsky and Rao, 1997; Rao and Kreutz-Delgado, 1999) of the following form

\[
\arg\min_{w \in \mathbb{R}^n} \sum_i ||w_i||^p \quad \text{subject to } Aw = y,
\]

which corresponds to a noiseless situation with \(f(w)\) being the \(l_p\) norm. The basic idea is to reduce the difficult original objective function to a series of re-weighted least squares problems. The earliest work of developing an IRLS on the \(l_1\) loss function was attributed to Schlossmacher (1973). Several later studies (Holland and Welsch, 1977; Scales and Gersztenkorn, 1988; Bube and Langan, 1997) noticed that the IRLS procedure could be applied to a large class of robust linear regression loss functions besides the absolute deviations loss. As to the convergence rate, IRLS has been shown to admit a linear rate generally (Lange et al., 2000; Bissantz et al., 2009). Under certain assumptions such as the \(k\)-sparsity and for a constrained programming form, Daubechies et al. (2010) prove that the IRLS procedure for \(l_p\) with \(p \in (0, 1]\) even admits a super-linear rate.

Quite often, the construction of a general IRLS procedure involves evaluation of the second-order derivative for the loss function and involves a smoothing procedure for a general non-smooth loss functions. Vogel and Oman (1996) and Lange et al. (2000) summarize many surrogate techniques for commonly used non-smooth functions. For theoretical investigation, we adopt an integral-based smoothing technique (Ghomi, 1997; Nesterov, 2005; Bissantz et al., 2009) to construct a smooth surrogate for the coercive convex \(f(w)\). We shall note that (Bissantz et al., 2009) considers only the convex function \(f(w)\) but here we will construct a different surrogate for the \(l_p\) loss function. This treatment considers refined structures of the \(l_p\) loss and yields better results and a faster convergence, as shown in the experiment section.

In fact, IRLS and many other famous procedures, such as expectation maximization (EM) (Dempster et al., 1977) and concave-convex procedure (CCCP) (Yuille et al., 2002), are examples of the minimization-majorization (MM) framework (Ortega and Rheinboldt, 1970; Lange et al., 2000). The MM framework is a general guideline for algorithm design. Using the entropy inequality, the EM algorithm transfers optimization of a loglikelihood function to expectation maximization of a posterior distribution. It is necessary to adaptively re-weight the whole formulation so that \(||A||_F \leq 1\). This re-weighting procedure leads to inconsistency in penalization weights and thus it is difficult to design fair comparisons with other methods.
function to a surrogate function. Also, the CCCP algorithm is built on a first-order optimization transfer function (Sriperumbudur and Lanckriet, 2009). The CCCP algorithm solves difference of convex functions (d.c.) programs as a sequence of convex programs. It works by introducing a first-order optimization transfer function (Sriperumbudur and Lanckriet, 2009) and then applying a minimization-majorization framework to derive algorithms. Although in theory any cost function with bounded Hessian matrix can be decomposed into a d.c. form (Yuille et al., 2002), such decomposition is often constructive and explicit reformulations are difficult to obtain for many important learning formulations such as the penalized $l_p$ problems. Motivated by the CCCP algorithm, in this work, we start by introducing a second-order optimization transfer method to deal with a family of penalized $l_p$ loss functions, and then we show how to use some approximation schemes to construct first-order practical algorithms.

3. Theoretical Derivations of Generic Iterative Procedure

This section is devoted to the theoretical derivations and properties of the proposed generic iterative procedure. We start by introducing different surrogate functions for the fidelity and the penalization terms, respectively. Afterwards, we develop the algorithm, prove its convergence, and discuss its convergence rate.

3.1 Notations and Preliminary

Unless otherwise specified, in this section, we always assume the coercive convex function $f(w)$ in Equation (1) is defined on a compact (i.e., bounded and closed) convex set $\mathcal{D} \subset \mathbb{R}^n$. Table 1 summarizes necessary symbols and notations. Throughout this paper, subscript $i$ denotes the $i$-th element, superscript $k$ denotes the $k$-th iteration.

3.2 Smooth Quadratic Surrogate for $f(w)$

The goal of this section is to construct a quadratic approximation of $f(w)$ from above. To achieve this, we proceed in two steps: 1) approximating $f(w)$ from above with a smooth surrogate $f_\mu(w)$; 2) approximating $f_\mu(w)$ by its quadratic approximation. Fortunately, these approximations are possible for coercive convex functions.

Definition 1 A nonnegative function $f_\mu(w)$, $\mu > 0$, is called a smooth surrogate for a nonnegative convex function $f(w)$, if $f_\mu(w)$ is continuously differentiable, strictly convex, and it satisfies $\lim_{\mu \to 0} |f_\mu(w) - f(w)| = 0$ for any $w \in \mathcal{D}$.

Taking the $f(w) = |w|$ as an example, some well-known smooth surrogates include the Huber function,

$$f_\mu(w) = \begin{cases} w^2/(2\mu) & \text{if } |w| \leq \mu \\ |w| - \mu/2 & \text{otherwise,} \end{cases}$$

or the logarithm form,

$$f_\mu(w) = \mu \log\left(\frac{e^{w/\mu} + e^{-w/\mu}}{2}\right),$$

3. Since $f(w)$ is coercive convex, restricting it to a compact set (e.g., by restricting to sublevel sets) yields simplified convergence proofs. We assume that a feasible solution exists in this compact set.
Table 1: Summary of notations.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>(m) or (n)</td>
<td>dimensionality</td>
</tr>
<tr>
<td>(\lambda)</td>
<td>regularization weight</td>
</tr>
<tr>
<td>(p) and (q)</td>
<td>vector or matrix norm identifier</td>
</tr>
<tr>
<td>(i)</td>
<td>(i)-th element of a vector</td>
</tr>
<tr>
<td>(k) and (l)</td>
<td>(k)-th or (l)-th iteration</td>
</tr>
<tr>
<td>(\mu) and (\epsilon)</td>
<td>small real values for dampening or ensuring numerical stability</td>
</tr>
<tr>
<td>(\nabla) and (\nabla^2)</td>
<td>gradient and Hessian matrix</td>
</tr>
<tr>
<td>(\xi)</td>
<td>convergence threshold</td>
</tr>
<tr>
<td>(\kappa)</td>
<td>a real-valued factor</td>
</tr>
<tr>
<td>(\gamma)</td>
<td>convergence rate</td>
</tr>
<tr>
<td>(\sigma_i)</td>
<td>(i)-th largest eigenvalue</td>
</tr>
<tr>
<td>(X \succ 0)</td>
<td>positive definiteness of matrix (X)</td>
</tr>
<tr>
<td>(L(w))</td>
<td>original loss function</td>
</tr>
<tr>
<td>(L_\epsilon(w))</td>
<td>(\epsilon) surrogate function for (L(w))</td>
</tr>
<tr>
<td>(Q_L)</td>
<td>the IRLS diagonal matrix for the loss term, positive definite</td>
</tr>
<tr>
<td>(Q_F)</td>
<td>the adjusted Hessian matrix for the penalization term, positive definite</td>
</tr>
<tr>
<td>(G(w</td>
<td>w^k))</td>
</tr>
<tr>
<td>(\hat{w})</td>
<td>solution to the original problem (Equation (1))</td>
</tr>
<tr>
<td>(\hat{w}_\mu)</td>
<td>solution to the surrogate problem (Equation (12))</td>
</tr>
<tr>
<td>(\hat{w}_{op})</td>
<td>solution to (G(w</td>
</tr>
<tr>
<td>(w^\infty)</td>
<td>a limit point of sequence ({w^k}_{k=1}^\infty)</td>
</tr>
</tbody>
</table>

or the square-root form,

\[
f_\mu(w) = \sqrt{w^2 + \mu^2}.
\]

For a general coercive convex function \(f(w)\), one way of constructing smooth surrogates from above is to convolve \(f(w)\) with a mollifier, whose explicit form can be found in Ghomi (1997). The following lemma (Ghomi, 1997; Bissantz et al., 2009) states the existence of a family of smooth surrogates \(f_\mu(w)\) from above for any coercive convex function \(f(w)\).

**Lemma 1** For any coercive convex function \(f(w)\) on a bounded closed convex set \(\mathcal{D} \subset \mathbb{R}^n\), there exists a family of smooth surrogates \(f_\mu(w)\) satisfying \(f_\mu(w) \geq f(w)\).

Proof: Please see Ghomi (1997); Bissantz et al. (2009).

Lemma 1 proves the existence of \(f_\mu(w)\), and we proceed to construct a quadratic approximation for \(f_\mu(w)\). By convolving with a mollifier, \(f_\mu(w)\) is infinitely often differentiable. Given a reference point \(w^k\), we construct the following quadratic approximation \(g_\mu(w)\) for \(f_\mu(w)\),

\[
g_\mu(w) = f_\mu(w^k) + \nabla f_\mu(w^k)(w - w^k) + \frac{(w - w^k)^T Q_F(w - w^k)}{2}.
\]

(2)

Here \(Q_F\) is an adjusted Hessian matrix of \(f_\mu(w)\). In this paper, we adopt the following adjustment strategy,

\[
Q_F = \kappa \nabla^2 f_\mu(w),
\]

(3)
where $\kappa$ is a factor to ensure that $Q_F \succ \nabla^2 f_\mu(w)$. Again, $X \succ 0$ denotes positive definiteness of $X$. Since $f_\mu(w)$ is strictly convex, both its Hessian $\nabla^2 f_\mu(w)$ and the adjusted Hessian $Q_F$ are positive definite. We summarize two important properties for this quadratic surrogate $g_\mu(w)$. First, $g_\mu(w)$ is an approximation of $f_\mu(w)$ from above (Lange et al., 2000) 4,

$$f_\mu(w) < g_\mu(w), \forall w \neq w^k;$$

second, the tangents coincide at $w^k$,

$$\nabla f_\mu(w)|_{w=w^k} = \nabla g_\mu(w)|_{w=w^k}.$$ 

3.3 IRLS Surrogate for $l_p$ Loss

Although the aforementioned smooth surrogate can also be used for the $l_p$ fidelity term in Equation (1), we will take a different surrogate based on the iterative re-weighted least squares (IRLS) method (Schlossmacher, 1973). This leads to an efficient algorithm which will be further explained in Section 4. By using a reference point $w^k$, the following IRLS $\epsilon$-surrogate can be constructed,

$$L_\epsilon(w) = \frac{1}{2}||Q_L(y - Aw)||^2 + (1 - \frac{p}{2})L(w^k),$$

where $\epsilon > 0$ is a small positive number, the IRLS diagonal matrix $Q_L$ is defined as

$$Q_L = \text{diag}(\rho_\epsilon(y - Aw^k)),$$

and the thresholding function $\rho_\epsilon(x)$ is defined as

$$\rho_\epsilon(x) = \begin{cases} |x|^{\frac{p}{2-2}} & \text{if } |x| \geq \epsilon, \\ \epsilon^{\frac{p}{2-2}} & \text{otherwise}. \end{cases}$$

The function $\rho_\epsilon(x)$ is mainly introduced for reducing numerical instability problems especially for $p < 2$. Other forms such as the Huber functions could also be used as $\rho_\epsilon(x)$. Also, this setting guarantees that $Q_L$ is positive definite which will be useful for the convergence analysis. The constant term $L(w^k)$ is provided to ensure that

$$\lim_{\epsilon \to 0} L_\epsilon(w^k) = L(w^k),$$

neglecting numerical issues. Also, we shall note that

$$L(w) < \lim_{\epsilon \to 0} L_\epsilon(w), \forall w \neq w^k,$$

and the tangents

$$\nabla L(w)|_{w=w^k} = \lim_{\epsilon \to 0} \nabla L_\epsilon(w)|_{w=w^k}.$$  

4. This work only considers bounded Hessian matrices. Equation (4) is valid by noting that both sides can be reformulated into forms of second-order derivatives. For unbounded Hessian matrices, where the quadratic approximation does not upper bound the original function, see Bissantz et al. (2009) for a further treatment.

5. For an exponential function $a^x$ with $x < 0$, if $a$ is small enough, the formulation may increase to infinity rapidly.
This setting has been used in previous studies (Holland and Welsch, 1977; Scales and Gersztenkorn, 1988; Chartrand and Yin, 2008) to investigate the IRLS for $l_p$ loss functions. A detailed proof for Equation (10) can be found in (Bube and Langan, 1997). The key idea is to construct a quadratic approximation of $L(w)$ from above and also maintain the same tangent direction on the tangent point. With these surrogates we introduce the following optimization transfer.

### 3.4 Optimization Transfer

Using a smooth surrogate function $f_\mu(w)$, we have the following surrogate formulation for Equation (1),

$$
\hat{w}_\mu = \arg\min_{w \in D} \{ L(w) + \lambda f_\mu(w) \}. \tag{12}
$$

Defining two cost functions for Equation (1) and Equation (12) respectively,

$$
C(w) := L(w) + \lambda f(w), \tag{13}
$$

$$
C_\mu(w) := L(w) + \lambda f_\mu(w), \tag{14}
$$

it is followed from the definition of $f_\mu(w)$ that $C_\mu(w)$ converges to $C(w)$ as $\mu \to 0$. Since $C(w)$ is convex and $D$ is compact, this point-wise convergence is also uniform. The following lemma shows that $\hat{w}_\mu$ approximates the solution $\hat{w}$ of Equation (1).

**Lemma 2** Assume the solution set for Equation (1) is $M$. As $\mu \to 0$, we have

$$
d(\hat{w}_\mu, M) := \inf_{\hat{w} \in M} ||\hat{w}_\mu - \hat{w}|| \to 0.
$$

Proof: Please see Theorem 1 of Bissantz et al. (2009). More details can be found in Borwein and Vanderwerff (1996).

We emphasize that Lemma 2 guarantees that the solution of a smoothed problem $C_\mu(w)$ converges to the solution of $C(w)$ as $\mu \to 0$. Based on Lemma 2, we turn to optimizing Equation (12). This is still challenging especially for $p = 1$ because the loss function $L(w)$ is non-differentiable. Inspired by the idea of optimization transfer (Lange et al., 2000), for any given $w^k$, we propose to optimize the following criterion,

$$
\hat{w}_{op} = \arg\min_{w \in D} \{ L_\epsilon(w) + \lambda g_\mu(w) \}. \tag{15}
$$

In the following, we first solve Equation (15) and construct the optimization transfer-based algorithm, and then we will prove that this optimization transfer indeed optimizes the criterion in Equation (12).

**Lemma 3** For a given $w^k$, the optimal solution to Equation (15) is given by

$$
\hat{w}_{op} = (A^T Q_L A + \lambda Q_F)^{-1} (A^T Q_L y + \lambda (Q_F w^k - \nabla f_\mu(w^k))). \tag{16}
$$

Proof: The optimal condition for Equation (15) is

$$
\nabla L_\epsilon(\hat{w}_{op}) + \lambda \nabla g_\mu(\hat{w}_{op}) = 0, \tag{17}
$$
which equals
\[ A^T Q_L (A\hat{w}_{op} - y) + \lambda (\nabla f_\mu(w^k) + Q_F(\hat{w}_{op} - w^k)) = 0, \]
and this can be further simplified as
\[ \hat{w}_{op} = (A^T Q_L A + \lambda Q_F)^{-1}(A^T Q_L y + \lambda (Q_F w^k - \nabla f_\mu(w^k))). \]

Starting from an arbitrary \( w_0 \), a sequence \( \{w^k\}_{k=0}^\infty \) can be constructed by iteratively evaluating \( w^k \) in Lemma 3. The following Algorithm 1 summarizes this idea and it will be called the Generic Iterative Procedure (GIP) for solving Equation (12). The parameters \( \mu \) and \( \epsilon \) are preset.

For a given penalization function \( f(w) \), if we can precompute the explicit form for \( Q_F \), the computational bottleneck of \( \hat{w}_{op} \) is on the matrix inversion. Moreover, the matrix inversion can be further circumvented by connecting the solution \( \hat{w}_{op} \) with a perturbed Tikhonov regression problem, so that highly efficient and scalable GIP can be constructed. We will leave the discussion of this connection to Section 4. In the following, we will adopt the concept of optimization transfer to prove the global convergence of Algorithm 1.

**Algorithm 1** A Generic Iterative Procedure (GIP) for Solving Equation (12)

**Input:** Data matrix \( A \), observation vector \( y \), factor \( \kappa \), stopping tolerance \( \xi \), smooth parameter \( \mu \), and dampening value \( \epsilon \)

**Initialize** \( k = 0 \) and \( w_0 \in \mathbb{R}^n \) to be an arbitrary non-zero vector.

**repeat**
- Evaluate \( Q_F \) based on Equation (3)
- Evaluate \( Q_L \) based on Equation (7)
- Evaluate \( \hat{w}_{op} \) based on Equation (16), \( k \leftarrow k + 1 \), and let \( w^k = \hat{w}_{op} \)

**until** \( ||w^k - w^{k-1}||_2 \leq \xi \) is true

### 3.5 Global Convergence

In this part, we explore the majorization-minimization (MM) framework (Lange et al., 2000), which can be viewed as the second layer of optimization transfer, to prove the global convergence of Algorithm 1. See the Section 2 for relationships between MM, EM, and CCCP. To minimize an objective function such as \( C_\mu(w) \), we may transfer optimization to a majorizing function \( G(w|w^k) \),

\[ G(w|w^k) = L_\epsilon(w) + \lambda g_\mu(w), \]

which is the cost function associated with Equation (15). According to the definition of \( L_\epsilon(w) \), \( G(w|w^k) \) satisfies the following condition

\[ G(w|w^k) \geq C_\mu(w), \forall w, \]  

and the equality holds only at \( w = w^k \). Then we can minimize \( G(w|w^k) \) and get a new \( w^{k+1} \), and this entails

\[ C_\mu(w^k) = G(w^k|w^k) > G(w^{k+1}|w^k) \geq C_\mu(w^{k+1}), \]

12
for any \( w^{k+1} \neq w^k \). This iterative process drives \( C_\mu(w) \) downhill for any \( w^{k+1} \neq w^k \).

Since \( C_\mu(w) \) is strictly convex, the global minimizer of Equation (12) is unique. Based on the aforementioned MM framework, the following theorem shows the global convergence of Algorithm 1.

**Theorem 1** For an arbitrary initialization point \( w^0 \in \mathcal{D} \), the sequence \( \{w^k\}_{k=0}^\infty \) generated by Algorithm 1 converges to \( \hat{w}_\mu \), the unique minimizer of Equation (12).

Proof: We separate the proof into three parts: 1) For any initialization vector \( w^0 \), the sequence \( \{w^k\}_{k=0}^\infty \) always drives the cost function \( C_\mu(w) \) downhill if \( w^{k+1} \neq w^k \) for any initialization point \( w^0 \). Since \( C_\mu(w) > 0 \) is lower bounded, the value sequence \( \{C_\mu(w^k)\}_{k=0}^\infty \) converges. Denoting the limit point of the value sequence as \( C_\mu(w^*) \), \( w^* \) is generally not unique because the initialization point is arbitrary. However, for any limit point \( w^* \), there always exists a subsequence \( \{w^{k_i}\}_{i=1}^\infty \) converging to it. Taking a limit point \( w^* \), we show that it is the only minimizer of \( G(w|w^*) \).

Step 1: At the \( k \)-th step, the reference point \( w^k \) is always used to construct the surrogate function \( L_\epsilon(w), g_\mu(w) \), as well as the objective optimization transfer function \( G(w|w^k) \). The \( w^{k+1} \) generated by Algorithm 1 minimizes \( G(w|w^k) \).

First we note that Equations (18) and (19) hold, and thus the sequence \( \{w^k\}_{k=0}^\infty \) always drives the cost function \( C_\mu(w) > 0 \) downhill if \( w^{k+1} \neq w^k \) for any initialization point \( w^0 \). Since \( C_\mu(w) > 0 \) is lower bounded, the value sequence \( \{C_\mu(w^k)\}_{k=0}^\infty \) converges. Denoting the limit point of the value sequence as \( C_\mu(w^*) \), \( w^* \) is generally not unique because the initialization point is arbitrary. However, for any limit point \( w^* \), there always exists a subsequence \( \{w^{k_i}\}_{i=1}^\infty \) converging to it. Taking a limit point \( w^* \), we show that it is the only minimizer of \( G(w|w^*) \).

Step 2: Taking any subsequence \( \{w^{k_i}\}_{i=1}^\infty \) with the limit point \( w^* \), we have

\[
\lim_{l \to \infty} C_\mu(w^{k_l+1}) \leq \lim_{l \to \infty} G(w^{k_l+1}|w^{k_l}) \leq \lim_{l \to \infty} G(w|w^{k_l}) = G(w|w^*).
\]

Considering the l.h.s. \( \lim_{l \to \infty} C_\mu(w^{k_l+1}) \geq C_\mu(w^*) = G(w^*|w^*) \), we conclude that

\[
G(w^*|w^*) \leq G(w|w^*), \forall w \in \mathcal{D}.
\]

This states that \( w^* \) is the only minimizer of \( G(w|w^*) \). For any \( k \) satisfying \( w^k = w^* \), we have \( w^{k+1} = w^k \).

Step 3: At this step, we will use the convexity of \( C_\mu(w) \) and \( G(w|w^k) \) to show that \( w^{k+1} = w^k \) if and only if \( w^k = \hat{w}_\mu \). First we note that for the unique minimizer \( \hat{w}_\mu \), we have the following sufficient and necessary condition (Rockafellar, 1970) for the strictly convex function \( C_\mu(w) \),

\[
\nabla w C_\mu(\hat{w}_\mu)(w - \hat{w}_\mu) \geq 0, \forall w \in \mathcal{D}.
\]

Considering Equation (11) and Equation (5), we always have

\[
\lim_{\epsilon \to 0} \nabla w G(w|w^k)|_{w=w^k} = \nabla w C_\mu(w)|_{w=w^k}.
\]

Since \( G(w|w^k) \) is convex, for the minimizer \( w^{k+1} \) of \( G(w|w^k) \), we also have the following sufficient and necessary condition (Rockafellar, 1970),

\[
\nabla w G(w^{k+1}|w^k)(w - w^{k+1}) \geq 0, \forall w \in \mathcal{D}.
\]

Combing Equation (20), Equation (21), and Equation (22), we conclude that \( w^{k+1} = w^k \) if and only if \( w^k = \hat{w}_\mu \) as follows.
• If \( w^k = \hat{w}_\mu \), then \( w^{k+1} = w^k \). Inserting \( \hat{w}_\mu = w^k \) into Equation (20) and using (21), we have
\[
\nabla w G(w^k | w^k)(w - w^k) \geq 0, \forall w \in D.
\]
This is a sufficient condition stating that \( w^k \) is a minimizer of \( G(w | w^k) \) and thus \( w^{k+1} = w^k \).

• If \( w^{k+1} = w^k \), then \( w^k = \hat{w}_\mu \). In this case, inserting \( w^{k+1} = w^k \) into Equation (22) and using (21), we have
\[
\nabla w C_\mu(w^k | w^k)(w - w^k) \geq 0, \forall w \in D.
\]
This is a sufficient condition stating that \( w^k \) is a minimizer of \( C_\mu(w) \) and thus \( w^k = \hat{w}_\mu \).

Putting these three steps together, we conclude that for any initialization point \( w^0 \), the sequence \( \{w^k\}_{k=0}^\infty \) generated by Algorithm 1 converges to \( \hat{w}_\mu \). ■

**Corollary 1** The solution found by Algorithm 1 converges to a solution of Equation (1) as \( \mu \to 0 \).

**Proof:** The solution found by Algorithm 1 is \( \hat{w}_\mu \), and the solution set for Equation (1) is \( M \). By Lemma 2, \( \hat{w}_\mu \) converges to \( \hat{w} \in M \). When Equation (1) is strictly convex, \( \hat{w} \) is unique and thus \( \hat{w}_\mu \) converges to \( \hat{w} \). ■

**3.6 Convergence Rate**

For convex penalizations, we have proved the convergence of Algorithm 1 and here we analyze its convergence rate. Assuming \( \{w^k\}_{k=0}^\infty \) is a sequence generated by Algorithm 1 with the limit point \( w^\infty \), we define the following iterative map \( M(w) \) as
\[
w^{k+1} = M(w^k).
\]

Following Lange et al. (2000), based on the following first-order Taylor expansion around \( w^\infty \)
\[
w^{k+1} = w^\infty + \nabla M(w^\infty)(w^k - w^\infty),
\]
we can approximately evaluate the convergence rate \( \gamma \) as
\[
\gamma := \lim_{k \to +\infty} \sup \left\{ \frac{\|w^{k+1} - w^\infty\|}{\|w^k - w^\infty\|} \leq \sigma_{\text{max}}(\nabla M(w^\infty)), \right. \tag{23}
\]
where \( \sigma_{\text{max}} \) denotes the largest eigenvalue. The following theorem shows that Algorithm 1 admits a linear convergence rate.

**Theorem 2** Algorithm 1 admits a linear convergence rate.

**Proof:** We start by evaluating \( \nabla M(w^\infty) \) and then we show that its maximum eigenvalue falls within the interval \([0, 1)\). By noting that \( \nabla G(w | w^k) = A^T Q_L (A w - y) + \lambda (\nabla f_\mu(w^k)) + Q_F(w - w^k) \) and \( \nabla^2 G = A^T Q_L A + \lambda Q_F \), we have
\[
w^{k+1} = \{\nabla^2 G\}^{-1}(A^T Q_L y + \lambda (Q_F w^k - \nabla f_\mu(w^k)))
= w^k - \{\nabla^2 G\}^{-1}(\nabla^2 G(w | w^k)w^k - A^T Q_L y - \lambda (Q_F w^k - \nabla f_\mu(w^k)))
= w^k - \{\nabla^2 G\}^{-1} \nabla G(w^k | w^k)
= w^k - \{\nabla^2 G\}^{-1} \nabla C_\mu(w^k).
\]
The last equality $\nabla G(w^k|w^k) = \nabla C_\mu(w^k)$ holds under the condition that $\epsilon \to 0$. The right hand side only involves $w^k$ and we can define

$$M(w^k) := w^k - \{\nabla^2 G\}^{-1}\nabla C_\mu(w^k).$$

Taking derivative w.r.t. $w^k$ for $M(w^k)$, we have the following formulation,

$$\nabla M(w^k) = I - \{\nabla^2 G\}^{-1}\nabla 2 C_\mu\{\nabla^2 G\} - \{\nabla^2 G\}^{-2}\nabla C_\mu\{\nabla^2 G\} \nabla w^k\{\nabla^2 G\}. $$

Letting $k \to \infty$ and using $\nabla C_\mu(w^\infty) = 0$, we have

$$\nabla M(w^\infty) = I - \{\nabla^2 G\}^{-1}\nabla^2 C_\mu(w^\infty).$$

(24)

Taking second-order derivatives on both sides of Equation (18), we have that

$$\nabla^2 G \geq \nabla^2 C_\mu(w).$$

The above equation is valid for any $w \in D$ and thus $\nabla^2 G - \nabla^2 C_\mu(w)$ is positive semidefinite, which implies that any eigenvalue $\sigma_i$ of $\{\nabla^2 G\}^{-1}\nabla^2 C_\mu(w)$ satisfying $\sigma_i \in (0, 1]$. Using Equation (24), it shows that the eigenvalues for $\nabla M(w^\infty)$ are within $[0, 1)$. At last, inserting this to Equation (23), we have

$$\gamma \leq \sigma_{\max}(\nabla M(w^\infty)) \in [0, 1).$$

(25)

This shows that Algorithm 1 admits a linear convergence rate.

4. Karmarkar’s Projective Scaling, Perturbed Tikhonov Regression, and Non-convexity

In this section we will develop practical efficient algorithms based on some constructive strategies. The constructions are only valid for some penalization functions. Especially, throughout this section, we do not require the convexity of the penalization functions, but we assume they are differentiable or each has at most one singular point such as $w = 0$ for $|w|$. For the last case, we simply omit the singular point in this section and leave further theoretical analysis to future work.

4.1 $l_q$ Family as an Example

If we can pre-compute the explicit form for the adjusted Hessian matrix $Q_F$, the GIP can be made very efficient. This is plausible for component-wise separable penalization functions $f(w) = \sum_i f_i(w_i)$, for which the resulted $Q_F$ is diagonal. As an important example, for the $l_q$ family

$$f(w) = \frac{1}{q} \sum_i |w_i|^q,$$

we can compute the gradient

$$\nabla f(w_i) = \text{sgn}(w_i)|w_i|^{q-1},$$

(26)
and its Hessian matrix
\[ \nabla^2 f(w_i) = (q - 1)\text{sgn}(w_i)\text{sgn}(w_i)|w_i|^{q-2} = (q - 1)|w_i|^{q-2}, \]
for any \( w_i \neq 0 \).

Considering the above Hessian matrix and taking \( \kappa = \frac{1}{q-1} \) in Equation (3), we can construct the following adjusted Hessian matrix with diagonal element
\[ Q_F(i,i) = |w_i|^{q-2}. \]
It is easy to verify that \( Q_F \succ \nabla^2 f_\mu(w) \) and \( Q_F \) is positive definite for any \( q \in [0,2] \).

Combining Equations (26) and (28), and the equality \( \text{sgn}(w_i) = w_i/|w_i| \), we have the following relationship
\[ Q_F w^k = \nabla f(w^k). \]
Inserting Equation (29) into Lemma 3 and we have the following simplification
\[ \hat{w}_{op} = (A^T Q_L A + \lambda Q_F)^{-1} A^T Q_L y. \]

In practice, to ensure numerical stability, a common strategy is to dampen each component of \( Q_F \) using a small number \( \delta > 0 \), and we adopt the following explicit form for \( Q_F \),
\[ Q_F(i,i) = (|w_i|^2 + \delta)^{\frac{q-2}{2}}. \]

**Remark (connections to the Karmarkar’s projective scaling strategy):** Equation (29) plays a key role in the above construction. It greatly simplifies the evaluation in Lemma 3 and we will explain its algorithmic advantages in the next section. In fact, Equation (29) is a form of the Karmarkar’s projective scaling strategy (Pino, 1989; Stone and Tovey, 1991; Bazaraa et al., 2009), which has been extensively used in interior point approach (Nash and Sofer, 1995; Bazaraa et al., 2009). In its original form, the Karmarkar’s Projective Scaling Strategy projects vectors from a simplex onto another simplex using a diagonal positive definite matrix. Using a similar idea, Equation (29) projects vectors \( (w) \) onto another gradient space (spanned by \( \nabla f(w) \)) using a positive definite matrix.

Besides the \( l_q \) family, here we show that we are able to construct Equation (29) for some other penalization functions as well. Assuming the gradient of the penalization function admits the following general form\(^6\),
\[ \nabla f(w^k) = \varphi(w^k)\nabla^2 f(w^k)w^k, \]
it follows from Equation (3) and Equation (29) that we require
\[ \varphi(w^k) = \begin{cases} 
\geq 1 & \text{if } \nabla^2 f(w^k) \text{ is positive definite}, \\
\leq -1 & \text{if } \nabla^2 f(w^k) \text{ is negative definite}. 
\end{cases} \]
This is consistent with the aforementioned \( l_q \) family where \( \varphi(w^k) = \frac{1}{q-1} \) and \( q \in [0,2] \).

Although still limited, Equation (32) and Equation (33) give a set of necessary conditions for constructing the aforementioned Karmarkar’s projective scaling strategy. For penalization functions satisfying these conditions, we proceed to apply Equation (30) to develop efficient algorithms.

\(^6\) This assumption is feasible for some component-wise separable penalization functions.
4.2 Perturbed Tikhonov Regression, Scalable GIP, and Sparsity

According to Equation (30), we can evaluate \( \hat{w}_{op} \) based on an explicit form which, however, still involves matrix inversion operations. Our aim is to leverage some efficient, possibly approximate, Tikhonov regression algorithms to partially circumvent the computationally demanding operations. The following theorem shows how to leverage the solution of a perturbed Tikhonov regression problem to solve \( \hat{w}_{op} \).

**Theorem 3** Given the following perturbed Tikhonov regression problem,
\[
\hat{x} = \arg\min_{x \in D} ||\bar{y} - (A + \Delta A)x||_2^2 + \lambda ||x||_2^2,
\tag{34}
\]
where \( \bar{y} = Q_{L}^{\frac{1}{2}}y \), \( \Delta A = Q_{L}^{\frac{1}{2}}AQ_F^{-\frac{1}{2}} - A \), we have
\[
\hat{w}_{op} = Q_F^{-\frac{1}{2}}\hat{x}.
\]

Proof: Temporarily let \( \Delta A = Q_{L}^{\frac{1}{2}}A - A \) and consider the following Tikhonov regression problem,
\[
\hat{\nu} = \arg\min_{\nu \in D} ||\bar{y} - (A + \Delta A)\nu||_2^2 + \lambda ||Q_{F}^{\frac{1}{2}}\nu||_2^2.
\tag{35}
\]
Its solution \( \hat{\nu} \) has an explicit form,
\[
\hat{\nu} = \{(Q_{L}^{\frac{1}{2}}A)^T(Q_{L}^{\frac{1}{2}}A) + \lambda Q_{F}^{\frac{1}{2}}Q_{F}^{\frac{1}{2}}\}^{-1}(Q_{L}^{\frac{1}{2}}A)^TQ_{L}^{\frac{1}{2}}y
= \{A^TQ_LA + \lambda Q_{F}\}^{-1}A^TQ_Ly = \hat{w}_{op}.
\]
Then we replace \( \nu = Q_{F}^{-\frac{1}{2}}x \) in Equation (35) and it follows that
\[
\Delta A = Q_{L}^{\frac{1}{2}}AQ_F^{-\frac{1}{2}} - A
\]
and \( \hat{w}_{op} = Q_{F}^{-\frac{1}{2}}\hat{x} \).

This theorem has two implications: developing scalable algorithms and understanding sparsity promoting property of the regularization terms. First, instead of directly evaluating more difficult Equation (1) with \( l_q \) penalizations, we may solve a series of perturbed Tikhonov regression problems. This enables us to leverage any efficient or possibly approximate ridge regression algorithms such as Turlach (2006), or any scalable least squares algorithms to approximately solve the original problem. Especially, for the \( l_2 \)-loss situations, we can construct highly efficient and scalable algorithms because \( Q_L \) is an identity matrix. The efficiency and scalability of the GIP are confirmed by the experiments.

Second, Theorem 3 shows an empirical connection between \( Q_F \) and the promoting of sparsity. Assuming a \( l_2 \) loss function (\( Q_L \) is an identity matrix) and a \( l_q \) penalization term with \( q \in [0, 2] \), initially we have
\[
w^0 = \arg\min_{w \in D} ||y - Aw||_2^2 + \lambda ||w||_2^2.
\]
Then we construct \( Q_F \) using Equation (31). It is easy to see that for \( q \in [0, 2) \) we have \( Q_F(i, i) \rightarrow 0 \) for \( w_{i}^0 \rightarrow 0 \). By the definition of \( \Delta A \) in Theorem 3 we note that an infinitely
large perturbation will be added to the $i$-th column of $A$. To achieve the minimization of the cost function, the next step solution,

$$
w^1 = \text{argmin}_{w \in D} ||y - (A + \Delta A)w||^2_2 + \lambda||w||^2_2,
$$

has to set the $i$-th column $w^1_i = 0$. On the other hand, for $q = 2$ the $Q_F$ is close to an identity matrix and thus no perturbation will be added. This empirical observation shows that $l_q$ regularization with $q \in [0, 2)$ promotes sparsity.

4.3 Non-convexity and Stationary Points

The projective scaling construction discussed in Section 4.1 is also applicable to many non-convex penalization functions, and we proceed to investigate the convergence behavior of the GIP for such non-convex situations. As we observed in experiments, for the non-convex situations, the GIP is not globally convergent but it still converges to a stationary point for a given initialization point. The following theorem shows that for differentiable but non-convex $C_\mu(w)$, the GIP converges to a stationary point.

**Theorem 4** For a given initialization point $w^0$, if $||\nabla^2 G(w)||_2$ is bounded, the sequence $\{w^k\}_{k=0}^\infty$ generated by Algorithm 1 converges to a stationary point of $C_\mu(w)$.

Proof: For a given initialization point $w^0$, we need to show that the sequence $\{w^k\}_{k=0}^\infty$ generated by Algorithm 1 leads to $\lim_{k \to \infty} \nabla C_\mu(w^k) = 0$. To simplify notations, we denote $G(w|w^k)$ as $G(w)$. First we note that Equations (18) and (19) still hold and the sequence $\{w^k\}_{k=0}^\infty$ entails $G(w^{k+1}) < G(w^k)$ for any $w^{k+1} \neq w^k$. Again, since $C_\mu(w) > 0$ is lower bounded, the value sequence $C_\mu(w^k)_{k=0}^\infty$ converges,

$$
\lim_{k \to \infty} |C_\mu(w^{k+1}) - C_\mu(w^k)| = 0.
$$

Using Equation (19) we have

$$
\lim_{k \to \infty} |G(w^{k+1}) - G(w^k)| = 0.
$$

Again, similar to the proof of Theorem 2, we have

$$
w^{k+1} = w^k - \{\nabla^2 G(w^k)\}^{-1} \nabla G(w^k).
$$

Taking the first-order Taylor approximation around $w^k$,

$$
G(w^{k+1}) = G(w^k) + \nabla G(w^k)^T (w^{k+1} - w^k) + O((w^{k+1} - w^k)^T \nabla^2 G(w^k)(w^{k+1} - w^k)),
$$

here we only keep the first two terms at the right hand side and omit any high-order expansions. By inserting Equation (38) into Equation (39), we have

$$
G(w^{k+1}) = G(w^k) - \nabla G(w^k)^T \{\nabla^2 G(w^k)\}^{-1} \nabla G(w^k)
= G(w^k) - ||\{\nabla^2 G(w^k)\}^{-1/2} \nabla G(w^k)||^2_2.
$$
Using \( \lim_{k \to 0} \nabla C_\mu(w^k) = \nabla G(w^k) \) and Equation (40),

\[
||\nabla C_\mu(w^k)||_2^2 = ||\{\nabla^2 G(w^k)\}^{1/2}\{\nabla^2 G(w^k)\}^{-1/2}\nabla G(w^k)||_2^2 \\
= ||\{\nabla^2 G(w^k)\}^{1/2}\{\nabla^2 G(w^k)\}^{-1/2}\nabla G(w^k)||_2^2 \\
= ||\{\nabla^2 G(w^k)\}^{1/2}\{\nabla^2 G(w^k)\}^{-1/2}||_2^2|G(w^{k+1}) - G(w^k)|. 
\]

(41)

Taking \( ||\{\nabla^2 G(w)\}^{1/2}||_2^2 \leq c \) where \( c \) is a bounded value, we have

\[
||\nabla C_\mu(w^k)||_2^2 \leq c||G(w^{k+1}) - G(w^k)||. 
\]

Using Equation (37), we conclude that \( ||\nabla C_\mu(w^k)||_2^2 \to 0 \) as \( k \to \infty \).

For non-convex situations, this discussion also raises two important questions: 1) how to choose a good initialization point; 2) is the GIP continuously dependent on an initialization point? We conduct extensive experiments and Figure 1 illustrates some global convergent and non-global convergent situations using the GIP on a sparse signal recovery experiment. The detailed experiment setting is reported in Section 6.1. Figure 1 (a) confirms that for convex formulations, the GIP is globally convergent; Figure 1 (b) and Figure 1 (c) show that for non-convex formulations, the GIP converges to a stationary point and the least squares solution (solution of Equation (1) without the regularization term) is a good initialization point in this setting\(^7\); Figure 1 (d) shows that all initialization points within a small ball of the least-squares solution appear to converge to the same solution in this setting. Since the GIP can be viewed as a series of perturbed Tikhonov regression problems, it is also of great interest to investigate its properties from a dynamic system perspective in future studies.

We observed in the experiments that the approximate GIP works well in many non-convex situations. Especially, we test \( l_q \) with \( q = 0 \) in our experiments and the results are encouraging. However, for these much more general situations such as \( l_0 \) (non-convex and discontinuous), the applicability of the GIP demands further theoretical justifications.

5. Applications to Low-rank Matrix Approximation

Given a measurement matrix \( Y \in \mathbb{R}^{m \times n} \) with some missing entries, we consider how to effectively decompose it into a product of two unknown low rank matrices \( U \in \mathbb{R}^{m \times r} \) and \( V \in \mathbb{R}^{n \times r} \). If we treat \( U \) as a set of basis and \( V \) as a set of coefficients\(^8\) and impose certain regularizations to the coefficients, this can be formulated as a regularized minimization problem

\[
\min_{U,V} \psi(U,V), 
\]

(43)

where

\[
\psi(U,V) = ||X \odot (Y - UV^T)||_p + \lambda||V||_q^q. 
\]

(44)

\( X \) contains indicators for the missing components and \( \odot \) is the Hadamard product. If \( Y_{ij} \) is missing, the corresponding \( X_{ij} \) is 0, otherwise \( X_{ij} = 1 \). \( \lambda \in \mathbb{R}^+ \) is a weight and \( ||\cdot||_p \) (or \( ||\cdot||_q \))

- Since the least squares solution is not robust, for general coefficients distributions with outliers of arbitrary magnitude, it might not be a desirable initialization point.
- \( U \) and \( V \) are interchangeable in meaning by noting that \( ||X \odot (Y - UV^T)||_p = ||X^T \odot (Y^T - VU^T)||_p \) in Equation (44).
is an "entry-wise" $p$ (or $q$) norm. Equation (43) appears repeatedly in many real-world applications such as structure-from-motion (Tomasi and Kanade, 1993; Eriksson and Hengel, 1992). We treat the $m \times n$ matrix as a vector of size $mn$ and use the $p$ (or $q$) norm for vectors. We require $p \in (0, 2], q \in (0, 2]$ in this paper. $p = 2$ is the Frobenius norm.

9. We treat the $m \times n$ matrix as a vector of size $mn$ and use the $p$ (or $q$) norm for vectors. We require $p \in (0, 2], q \in (0, 2]$ in this paper. $p = 2$ is the Frobenius norm.
Optimization Transfer for lp Loss with Convex Penalizations (2010), photometric stereo (Hayakawa, 1994), and recommendation systems (Koren et al., 2009). However, for general p and q, especially for those non-differentiable regularization terms, how to effectively and efficiently solve Equation (43) is still challenging. In this part, we show that the proposed GIP is applicable to this matrix approximation problem.

5.1 A Reformulation of the Low-rank Matrix Approximation

Our starting point is to employ the well-known Wiberg like procedure (Wiberg, 1976). In fact, by letting \( U = [u_1, ..., u_m]^T \), \( V = [v_1, ..., v_n]^T \), and defining \( u := [u_1^T, ..., u_m^T]^T \), \( v := [v_1^T, ..., v_n^T]^T \), Equation (43) is equivalent to the following least squares-like problem:

\[
\min_{u,v} \phi(u, v),
\]

where

\[
\phi(u, v) = ||\tilde{y} - Fu||_p^p + \lambda ||v||_q^q = ||\tilde{y} - Gv||_p^p + \lambda ||v||_q^q.
\]

Letting \( \tilde{y} \) and \( x \) be two vectors containing all components of \( Y \) and \( X \) in a lexical order of \( i \) and \( j \) respectively, \( \tilde{y} := \text{diag}(x)\tilde{y}, F := \text{diag}(x)(V^T \otimes I_n), G := \text{diag}(x)(I_n \otimes U) \) where \( \otimes \) is the Kronecker product. The norms in Equation (46) are vector norms. After deleting missing entries in \( \tilde{y} \), \( F \) and \( G \), the length of vector \( \tilde{y} \) is \( mn - s \) where \( s \) is the number of missing components in \( Y \).

By eliminating \( X \), Equation (45) has a much simpler form than Equation (43). However, it still involves simultaneous optimization with two unknown vectors \( u \) and \( v \). To tackle this, we adopt the general framework of Ruhe and Wedin (1980) and introduce a first-order approximation similar to the Wiberg procedure (Wiberg, 1976; Okatani and Deguchi, 2007). As shown in (Okatani and Deguchi, 2007), without any regularization term, the Wiberg procedure can effectively compute Equation (45) for \( p = 2 \).

5.2 Linearization

In order to perform simultaneous optimization, first we have to reduce the number of unknown variables. This can be done by computing the analytic solution of one variable and then substituting this analytic solution into an equivalent formulation of \( \phi(u, v) \). For a fixed \( v \), we can compute the analytic solution \( \hat{u}(v) \) for \( u \) by solving

\[
\hat{u}(v) = \min_u ||\tilde{y} - Fu||_p^p + \lambda ||v||_q^q,
\]

which is equivalent to

\[
\hat{u}(v) = \min_u ||\tilde{y} - Fu||_p^p.
\]

Often, we can obtain (approximately) analytic solutions for Equation (48) especially for \( p = 2 \) (Wiberg, 1976) and \( p = 1 \) (Eriksson and Hengel, 2010). By substituting this analytic solution into Equation (45) again, we have

\[
\hat{v} = \min_v \phi(\hat{u}(v), v) = \min_v ||\tilde{y} - G(\hat{u}(v))v||_p^p + \lambda ||v||_q^q
\]

Generally, Equation (49) is a nonlinear least-squares-like problem and it is difficult to solve. We turn to use a first-order approximation for \( \tau(v) = \tilde{y} - G(\hat{u}(v))v \) and \( \gamma(v) = v \) around a
known $v_k$

$$
\tau(v) = \tau(v_k) + \tau'(v_k)(w - v_k)
$$

$$
\gamma(v) = \gamma(v_k) + \gamma'(v_k)(w - v_k) = w
$$

and the resulted problem is now a linear least-squares-like problem with an unknown vector $w$

$$
\hat{w} = \min_w ||\tilde{y} - G(\hat{u}(v_k))v_k + \tau'(v_k)(w - v_k)||^p_p + \lambda||w||^q_q.
$$

(50)

Finally, Equation (50) can be further simplified as

$$
\hat{w} = \min_w ||y - Aw||^p_p + \lambda f(w),
$$

(51)

where $y = G(\hat{u}(v_k))v_k + \tau'(v_k)v_k - \tilde{y}$ and $A = \tau'(v_k)$ are known for fixed $v_k$, $w$ is the unknown coefficient vector and $f(w) = ||w||^q_q$. Without the regularization term, the whole process is known as the Wiberg procedure. By using the proposed GIP procedure, we can effectively solve Equation (51), and the update is given by $v_{k+1} = v_k + \hat{w}$.

For each iteration, the GIP involves solving a ridge regression problem with data matrix of size $M \times N$, which incurs a cost of $O(MN^2)$. The size of $A$ in Equation (51) is $T \times nr$ where $T$ is the number of non-missing entries, $n$ is the number of columns of $Y$, and $r$ is the rank. So each iteration is dominated by $O(Tn^2r^2)$.

6. Experiments

In this part, we design several experiments, including simulated signal recovery, face image classification, sparse dictionary learning-based image denoising, simulated low-rank matrix approximation, and structure-from-motion to verify the correctness and efficiency of the proposed optimization transfer algorithms. The experiments are mainly carried out with $l_p$ ($p \in \{0.5, 1, 2\}$) loss functions penalized by $f(w)$ as $l_q$ ($q \in \{0, 0.5, 1, 2\}$) regularization terms. For the proposed optimization transfer-based method using $l_p$ loss and $l_q$ regularization, we denote it as GIP($l_p + l_q$). For GIP($l_2 + l_q$) algorithms, there is no need to perform explicit matrix inversion and we simply adopt the Matlab’s built-in ridge function to solve the perturbed ridge regression problem. The $\epsilon$ is fixed to be 1e-10 for the GIP. The default stopping tolerance is 1e-5 for all methods unless otherwise stated.

We mainly compare the GIP with other state-of-the-art methods in three categories, lasso, $l_0$ penalized least squares problems, and penalized low-rank matrix decomposition-based problems. For lasso-related applications, we use CVX solvers\(^{10}\) (Grant and Boyd, 2009) as a baseline and compare GIP($l_2 + l_1$) with GPSR\(^{11}\) (Figueiredo et al., 2007), FPC-AS\(^{12}\) (Wen et al., 2009), the unconstrained programming version of NESTA\(^{13}\) (Becker et al., 2009) which will be denoted as NESTA-UP. All these methods use an unconstrained programming formulation and we require the same penalization weight and stopping tolerance for all methods in each experiment. It shall be noted that FPC-AS is only applicable to under-determined systems and other methods are generally applicable. For $l_0$

\(^{10}\) http://cvxr.com/cvx/
\(^{11}\) http://www.lx.it.pt/~mtf/GPSR/
\(^{12}\) http://www.caam.rice.edu/optimization/L1/FPC_AS/
\(^{13}\) http://www-stat.stanford.edu/~candes/nesta/
penalized least squares problems, we compare GIP($l_2 + l_0$) with SL0\textsuperscript{14} (Mohimani et al., 2009), ISD strategy\textsuperscript{15} (Wang and Yin, 2010), and re-weighted $l_1$ (RL1)\textsuperscript{16} strategy (Candés et al., 2008). Since $l_0$ based methods often have different parameters, we use default settings from their original implementations. For matrix decomposition-based structure-from-motion problems, besides comparisons with GPSR and NESTA-UP, we also compare GIP($l_2 + l_1$), GIP($l_2 + l_0$) with several well-known methods such as the Wiberg (Wiberg, 1976), $l_1$ Wiberg (Eriksson and Hengel, 2010), and the damped Newton method\textsuperscript{17}. The penalization weights and stopping tolerances are application-related, and we use the same value for all methods as long as they use the same cost functions; for other parameters, such as smoothness parameters for NESTA-UP and SL0, we use their default values from authors’ original implementations. It shall be noted that we will not perform any form of thresholding for GIP($l_p + l_q$) algorithms in all experiments. An implementation of our method can be downloaded\textsuperscript{18}.

### 6.1 Synthetic Sparse Signal Recovery

The aim of the experiments in this section is to empirically obtain a better understanding of the proposed method. We design three sets of experiments. The first experiment is to compare reconstruction accuracies of different lasso-based algorithms such as FPC-AS, GPSR, NESTA-UP, and GIP($l_2 + l_1$), based on a statistical analysis; the second experiment is to illustrate the performances of $l_0$-based algorithms such as SL0, ISD, and RL1; at last, we also show the performances of various GIP($l_p + l_q$) algorithms with different $p$ and $q$.

A dataset is generated as follows: first, we generate a one dimensional discrete signal of length 200 with coefficients all zeros. Then we put 15 spikes to produce a sparse signal $w$. Next we make 80 observations by using a random projection matrix. Finally Gaussian noise is added to get the noisy observations. This procedure is repeated to generate 1000 datasets\textsuperscript{19}.

We use a small penalization weight ($1e-5$) to test signal recovery ability of various methods. The reconstruction error is defined as $l_2$ norm of the difference between recovered and original signals. Figure 2 reports performances of compared methods on a dataset. It shows that FPC-AS has the highest accuracy and NESTA-UP has the lowest accuracy; GIP($l_2 + l_1$) has better accuracy than GPSR. The mean errors for compared methods are shown in Figure 3. Especially, out of 1000 datasets, GIP($l_2 + l_1$) achieves lower reconstruction errors than FPC-AS, GPSR, and NESTA-UP on 937, 984, 994 datasets respectively. It confirms that GIP($l_2 + l_1$) has better accuracies than GPSR and NESTA-UP.

Besides lasso, we also give illustrative results on $l_0$ penalized least squares problems. Figure 4 shows performances of GIP($l_p + l_0$), SL0, ISD, and RL1. It shall be noticed that ISD is only applicable to under-determined systems. Also, RL1 involves a $l_1$ minimizer for

\textsuperscript{14} http://ee.sharif.ir/~SLzero/
\textsuperscript{15} http://www.caam.rice.edu/~optimization/L1/ISD/
\textsuperscript{16} We implemented a RL1 according to the original paper (Candés et al., 2008).
\textsuperscript{17} The damped Newton method uses the Levenberg-Marquardt algorithm to solve the nonlinear least squares problem.
\textsuperscript{18} http://www.cs.siu.edu/~hbzhou/gip/
\textsuperscript{19} Since we use a small value for the penalization parameter, to avoid datasets which are unsuitable for all compared methods, we use FPC-AS as the baseline and reject any dataset where FPC-AS’s recovery error is greater than 0.2.
Figure 2: Illustration of the sparse signal recovery example using GIP($l_p + l_1$), FPC-AS, GPSR, and NESTA-UP on the same dataset. Ideally, the original and recovered points should overlap. The circles “◦” represent the original signals and the diamonds “⋄” denote the recovered signals.

Each iteration and thus it is computationally demanding. GIP($l_p + l_0$) and SL0 are generally applicable and GIP($l_p + l_0$) outperforms SL0. Using the same setting, we also test several combinations of $p$ and $q$ using the GIP($l_p + l_q$) method, and the detailed results are reported in supplementary document.

6.2 Face Image Classification

In this section, we conduct experiments to compare different methods’ efficiency and scalability. Recent research suggests that sparse representation-based classification method is promising in face recognition problems (Wright et al., 2009). Since this application involves high dimensional datasets, it is of interest to investigate the performance of our proposed algorithms on these problems. We implement a sparse representation classification framework

Optimization Transfer for \( l_p \) Loss with Convex Penalizations

Figure 3: Illustration of mean reconstruction errors for FPC-AS, GPSR, NESTA-UP, and GIP\((l_2 + l_1)\) over 1000 runs.

as in Wright et al. (2009) and compare two categories of algorithms, lasso-based methods such as CVX (lasso), FPC-AS, GPSR, NESTA-UP, GIP\((l_2 + l_1)\), and \( l_0 \) penalization-based methods such as SL0, ISD, GIP\((l_2 + l_0)\). RL1 is computational demanding and thus omitted.

Each column of the data matrix \( A \) represents concatenated pixels of a face image. For a given dataset, we perform the same leave-one-out test for all five methods. For each run, we first uniformly sample an image as a testing example \( y \) and use all other images as training examples. Then we compute the vector \( w \) using each method. At last, we adopt the minimum distance decision rule (Wright et al., 2009) to predict the label for the testing example \( y \). This process is repeated 100 times and finally we compare the accuracy of testing and the time costs\(^{21}\) (averaged over 100 runs).

First we perform experiments on the EYaleB database (Georghiades et al., 2001; Lee et al., 2005), which includes 38 individuals and around 64 near frontal images under different illuminations per individual. All 2414 images are down-sampled to five different dimensions 1024, 529, 256, 100, 49, respectively. There is no significant difference among all compared methods in recognition performance on this dataset and all methods are almost error-free\(^{22}\). Figure 5 shows the averaged time costs for computing the coefficients for each testing example for different dimensions. For a down-sampled image of 32 × 32 pixels, GIP\((l_2 + l_1)\) needs about 2 seconds to compute the coefficient over a set of 2413 training images while CVX requires more than 478 seconds. In conclusion, GPSR, GIP\((l_2 + l_1)\) and GIP\((l_2 + l_0)\) are scalable w.r.t. both dimension and sample size, and all of them are orders of magnitude faster than standard CVX solvers on relatively large datasets.

\(^{21}\) All reported time costs are based on the same computer with CPU 3 GHz, Memory 16 GB.
\(^{22}\) Please see supplementary materials for detailed results. http://www.cs.siu.edu/~hbzhou/gip/
Figure 4: Illustration of the sparse signal recovery using $\text{GIP}(l_p + l_0)$, SL0, ISD, and RL1 on the same dataset. Ideally, the original and recovered points should overlap. The circles “○” represent the original signals and the diamonds “□” denote the recovered signals. The ISD is only applicable to under-determined systems and also not scalable as shown in the next experiment.

We further conduct experiments on a larger database. The CMU-PIE database (Sim et al., 2002) has 41,368 images of 68 people, with each person under 13 different poses, 43 different illumination conditions, and four different expressions. In this experiment, we use five near-front poses (C05, C07, C09, C27, C29) and all the images have different illuminations and expressions. There are 170 images for each individual and totally we have 11,554 images. Similarly to the previous test, all images are down-sampled to five different dimensions 1024, 529, 256, 100, 49, respectively and 100 runs of leave-one-out tests are performed. Based on results from the EYaleB database, we omit some time-consuming methods and choose to only compare the efficiency of FPC-AS, GPSR, NESTA-UP, and the GIP. Figure 6 shows the averaged time costs for computing the coefficients for each testing example for different dimensions. For a testing image of $32 \times 32$ pixels, $\text{GIP}(l_2 + l_1)$ needs about 39 seconds to compute the coefficient over a set of 11,554 training images while
Figure 5: Comparison of the efficiency of compared methods on the EYaleB database. To compute the coefficient $w$ for a $32 \times 32$ image over a set of 2413 training images, the averaged (over 100 runs) time costs for CVX (lasso), FPC-AS, GPSR, NESTA-UP, SL0, ISD, GIP($l_2 + l_1$), and GIP($l_2 + l_0$) are 478.29, 7.05, 4.59, 8.82, 21.44, 25.78, 2.38, 1.92 seconds, respectively. Among all these methods, it is noted that GPSR, NESTA-UP, GIP($l_2 + l_1$), and GIP($l_2 + l_0$) have the best scalability.

FPC-AS requires more than 70 seconds. In conclusion, the GIP is more efficient than GPSR and FPC-AS on this relatively large dataset.

6.3 Dictionary Learning for Image Denoising

In this section, we report the compared results on sparse dictionary learning-based image denoising. Since the seminal work of Olshausen and Field (1996), dictionary-based image denoising has attracted much attention (Chen et al., 2001; Portilla et al., 2003; Elad and Aharon, 2006; Mairal et al., 2008). It works by learning a sparse dictionary from a large set of redundant images patches, and then representing the image (patches) using the learned items. Given a set of image patches $y \in Y$, the dictionary learning aims at finding a set of representative atoms $D$. To this end, initialized with a random set of patches $D^0$, the learning procedure alternatively updates the following two equations,

$$\hat{w} = \min_w ||y - D^i w||_2^2 + \lambda ||w||_1,$$

$$D^{i+1} \leftarrow \min_D ||Y - DW||_2^2,$$

27
where $W = [w_1, ..., w_d]$ and $d$ is the number of atoms. We follow the work of Elad and Aharon (2006) and consider additive zero-mean white and homogeneous Gaussian noise in this experiment. We adopt different lasso or $l_0$ penalized least-squares solvers to solve Equation (52), and we also compare with standard methods such as Wavelet-based (Bayesian soft thresholding) (Chang et al., 2000) and orthogonal matching pursuit (OMP)-based (Chen et al., 2001) image denoising methods. Besides the aforementioned methods, there are also other advances in image denoising, especially the field-of-expert (FOE) model (Roth and Black, 2005) and BM3D method (Dabov et al., 2007). We shall note that both FOE and BM3D require estimations of true noise models as inputs, which is difficult in practice. Additionally, FOE relies on the training datasets and it is also much less efficient than BM3D method.

We conduct two sets of experiments using color “Lena” and “Flower” images respectively. We first resize two color images to 256 x 256 pixels and then add some heavy Gaussian noise (zero mean, standard deviation 0.1) to each channel. For a given color image, we process the R, G, B channels independently because noise distributions are different for each channel. After extracting 5760 patches of 12 x 12 pixels, we train a dictionary of 288 atoms
Table 2: Comparison of PSNRs for denoised images using different methods. 10 bases are used for OMP and a two-layer “haar” wavelet decomposition is used for Wavelet-based method.

<table>
<thead>
<tr>
<th></th>
<th>Wavelet-based</th>
<th>OMP-based</th>
<th>NESTA-UP</th>
<th>GPSR</th>
<th>GIP($l_1$)</th>
<th>GIP($l_0$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Color Lena</td>
<td>25.18 dB</td>
<td>25.46 dB</td>
<td>25.21 dB</td>
<td>25.56 dB</td>
<td>25.56 dB</td>
<td>25.96 dB</td>
</tr>
</tbody>
</table>

and then perform sparse coding-based image denoising using Equation (52). The sparse coding for each patch is efficient (because each path is only of length 144) but the large amount of patches make this task computationally intensive. Without any parallelization\(^\text{23}\), we mainly choose to compare several efficient methods, GIP($l_2 + l_1$), GPSR, NESTA-UP, GIP($l_2 + l_0$), as well as two standard methods, OMP and Wavelet-based (Bayesian soft thresholding)-based image denoising. We adopt the standard peak signal-to-noise (PSNR) the performance metric and higher PSNRs represent better performances. All exogenous parameters including noise and stopping tolerance (1e-3) are fixed for all methods. The weighting parameter is the same for all lasso-based methods; the optimal number of bases for OMP is obtained by cross-validation; the best performance of Wavelet-based is obtained by testing different wavelets and decomposition layers. We also run SL0 and ISD on this application but no satisfying results have been observed even with tuning of their parameters.

Figure 7 illustrates some patches from the R-channel of the noisy “Lena” image, as well as some learned dictionary atoms using GIP($l_2 + l_0$). Table 2 reports SNRs of the denoised images for compared methods. Figure 8 and Figure 9 illustrate the original images, as well as some denoised images. As it shown, GIP achieves the highest SNR values comparing with GPSR, NESTA-UP, OMP, and Wavelet-based methods. Using an estimation of noise model of zero mean and standard deviation 0.06, BM3D achieves 24.28 dB for the same noisy “Lena” image. Again, we emphasize that estimation of the true noise model is often implausible in practice.

### 6.4 Experiments on Low-rank Matrix Approximation

From the above experiments, we conclude that FPC-AS and ISD are only applicable to under-determined systems and thus have limited applicability; the above experiments also show that CVX, GPSR, NESTA-UP, and the GIP are generally applicable but CVX is not scalable; the GIP has better efficiency than GPSR and NESTA-UP. The experiments designed in this section have high requirements on both accuracy and scalability.

In this part, we compare GIP($l_2 + l_1$), GPSR, and NESTA-UP on some penalized missing data low-rank matrix approximation problems. We also compare the GIP with other well-known missing data low-rank approximation methods such as Wiberg, $l_1$ Wiberg, and the damped Newton method\(^\text{24}\). Specifically, the $l_1$ Wiberg solves the $l_1$ loss function with-

---

\(^{23}\) We shall note that parallelization may greatly reduce overall time costs for this application.

\(^{24}\) The damped Newton method uses the Levenberg-Marquardt algorithm to solve the nonlinear least squares problem.
Zhou and Cheng

Figure 7: Illustration of patches from the R-channel of the noisy “Lena” image, and some trained dictionary atoms by GIP($l_2 + l_0$).

out regularization terms; the damped Newton method solves the $l_2$ loss function with a regularization term of $\lambda_1||V||_2 + \lambda_2||U||_2$.

6.4.1 Simulated Low-rank Matrix Approximation Problems

For the synthetic tests a set of random matrices $U \in \mathbb{R}^{50 \times 3}, V \in \mathbb{R}^{3 \times 30}$ are generated. The elements of $U$, $V$ are drawn from a uniform distribution $[-2, 2], [-3, 3]$ respectively. For each pair of $(U, V)$, a measurement matrix $Y = UV$ is computed. In addition, we add uniformly distributed noise over $[-1,1]$ to all elements of $Y$. Assuming band-diagonal non-missing components, we remove 50% of entries as missing components.

For this simulated dataset, we compare the performance and efficiency of three algorithms: GIP($l_2 + l_1$), Wiberg, and $l_1$ Wiberg. Figure 10 shows the fitting errors between $U, V$ and estimated $\hat{U}$ and $\hat{V}$, for each method. While the $l_1$ Wiberg and the GIP($l_2 + l_1$) have similar performances in recovering $\hat{V}$, the GIP($l_2 + l_1$) reconstructs a better $\hat{U}$ than the $l_1$ Wiberg. Also, as to the efficiency, the time costs are 0.10 seconds, 0.18 seconds, and 128.26 seconds for the Wiberg, GIP($l_2 + l_1$), and $l_1$ Wiberg respectively.

For matrix decomposition with missing components, the Wiberg method is computationally efficient but sensitive to heavy noise and outliers. On the other hand, $l_1$ Wiberg may have good performance in combating noise and outliers but it is computationally demanding for real-world applications. The GIP($l_2 + l_1$) method has both advantages: it has the same computational complexity as the Wiberg method; it can deal with heavy noise and outliers by using well-designed penalization terms.
6.4.2 Structure from Motion

We use the standard Dinosaur sequence which contains projections of 319 points tracked over 36 views. 72% of elements are missing for the measurement matrix and Figure 11 (a) shows the originally measured tracks. The \( l_1 \) Wiberg does not scale well for this dataset and we will mainly compare with the damped Newton method. Figure 11 (b) is the best result over 1000 runs obtained by Buchanan and Fitzgibbon (2005) using the damped Newton method to solve \( l_2 \) loss with a penalization term \( \lambda_1 ||U||_2 + \lambda_2 ||V||_2 \). Figure 12 shows the best result obtained by the GIP(\( l_2 + l_1 \)) over 50 runs with random initialization vectors. In Figure 12, (a) shows the reconstructed tracks, (b) is the recovered camera matrix and (c) is the recovered structure matrix for the dinosaur. Since the Dinosaur was placed on a

---

25. Available from http://www.robots.ox.ac.uk/~vgg
Zhou and Cheng

Figure 9: Illustration of the original “Flower” image, noisy image, denoised image by Wavelet Bayesian soft thresholding-based method, and denoised image by GIP($l_2 + l_0$).

turnable table, all tracks should be closed and circular. The GIP($l_2 + l_1$) performs better than the damped Newton method in Buchanan and Fitzgibbon (2005).

Since this application requires high accuracy and scalability, it serves as a natural choice for comparing various methods. Figure 13 shows convergence curves of GPSR, NESTA-UP, and GIP($l_2 + l_1$) on the Dinosaur sequence without outliers. It is clear that GIP($l_2 + l_1$) converges to a smaller cost. Figure 14 shows the recovered tracks by GPSR and we conclude that the GIP outperform GPSR and NESTA-UP on this problem. Since the resulted penalization problems are over-determined, FPC-AS is inapplicable.

To further simulate the presence of heavy noise and outliers, a uniformly distributed noise over [-50,50] is added to 10% of the elements of the measurement matrix. Figure 15 (a) shows the tracks from the measurement matrix and many tracks are heavily deteriorated.
Figure 10: The fitting errors between the original $U, V$ and recovered $\hat{U}, \hat{V}$. The horizontal axis denotes the index of the elements (column first) in matrices and the vertical axis shows the values of $U - \hat{U}$ or $V - \hat{V}$ for each element. As shown, the Wiberg method is sensitive to noise, while the $l_1$ Wiberg and GIP($l_2 + l_1$) are more robust. The penalization is on matrix $V$.

We run the GIP($l_2 + l_1$) and GIP($l_2 + l_0$) for 50 runs with random initialization vectors and the best results are reported in Figure 15 (b), (c), (d), respectively. For GIP($l_2 + l_1$), the averaged time for each run is about 60 seconds to obtain the reported results.

We shall note that trace norm-like penalization terms such as those in Buchanan and Fitzgibbon (2005) or the Wiberg do not perform well on similar SFM problems with missing components, and Mitra et al. (2010) imposes various orthographic constraints to the camera matrix. In this experiment, we show that simply adding $l_1$ regularization may work well on these problems. Moreover, in the presence of heavy noise and outliers, contrary to the $l_1$ Wiberg method, the optimization transfer-based methods (especially for $l_2$ loss function)
are scalable and thus can effectively solve corresponding regularized matrix decomposition problems.

7. Discussions and Conclusion

In this paper, we propose a generic iterative procedure (GIP) for effective and efficient optimization of a penalized $l_p$ learning problem. We conclude that the GIP is practically applicable to a large family of convex and even non-convex penalization functions. For coercive convex regularization functions, we prove the global convergence of the GIP and characterize its convergence rate; for other regularization functions especially the $l_q$ with $q \in [0, 2]$, we show how to adopt some approximation schemes to construct efficient GIP. We conduct extensive experiments to verify the effectiveness, genericness, and applicability of the proposed method. Again, for the effectiveness, we have tested accuracy, efficiency, and scalability of the proposed method; for the genericness, we have shown its applicability to some functions from the $l_p$ family; at last, the applicability has been verified through various under-determination and over-determination problems.

For either theoretical exploration or practical application purposes, there are several topics demanding further investigations. First, exactly solving Equation (16) is impractical for very large problems, and thus it is natural to investigate the GIP’s robustness w.r.t. the inexact ridge regression or least squares solvers. This could be done by leveraging studies on interior-point methods (Boyd and Vandenberghe, 2004) where prematurely terminated conjugate gradients are often used in solving the Newton step. As one reviewer pointed
out, purposefully solving the inner problem inaccurately (but quickly) is advantageous if we can establish some robustness results on this. Second, the fact that Equation (1) can be decomposed into a series of perturbed Tikhonov regression problems offers a novel perspective for understanding many penalized $l_p$ learning problems. This observation indicates that the effect of various regularization terms and regularization weights may be understood by using some perturbation analysis techniques (Ghaoui and Lebret, 1997; Zhou and Cheng, 2010). At last, a characterization of function classes which are suitable for iterative re-weighted least squares procedures would be useful for future algorithmic designs. The unifying role of iterative generalized least squares in statistical algorithms was first noticed by Pino (1989). Stone and Tovey (1991) carefully studied the well-known simplex method and the Karmarkar’s projective scaling method in linear programs, and their study concludes that these methods are essentially equivalent to the iterative re-weighted least squares. However, there still lacks an explicit characterization of functional classes on this problem.

Acknowledgments

We greatly appreciate the efforts of anonymous reviewers and their helpful comments and suggestions for improving the manuscript.
Figure 13: Convergence curves of GPSR, NESTA-UP, and GIP($l_2 + l_1$) on the Dinosaur sequence without outliers. We use the same settings, tolerance 1e-8 and penalization weight 0.1, for all methods.

References


Figure 14: Experiments on the Dinosaur sequence without outliers using GPSR. (a) shows the reconstructed tracks, (b) is the recovered camera matrix and (c) is the recovered structure matrix for the Dinosaur. For illustration purpose, the matrices are rescaled.


Figure 15: Experiments on the Dinosaur sequence with heavy noise and outliers. (a) shows
the tracks from the measurement matrix and many tracks are heavily deteriorated. Figures (b), (c) and (d) report the best results over 50 runs for the
Wiberg, GIP($l_2 + l_1$) and GIP($l_2 + l_0$), respectively. The averaged time for each
run is about 60 seconds to obtain the reported results. As shown, in the presence
of heavy noise and outliers, the optimization transfer-based methods have bet-
ter performance than the Wiberg. The GIP($l_2 + l_0$) appears to perform slightly
better than the GIP($l_2 + l_1$).


I. Daubechies, R. Devore, M. Fornasier, and C. S. Güntürk. Iteratively reweighted least

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data
via the em algorithm. Journal of the Royal Statistical Society Series B (Methodological),

P. S. Dhillon, D. Foster, and L. Ungar. Feature selection using multiple streams. In Pro-

D. L. Donoho and Y. Tsaig. Fast solution of $l_1$-norm minimization problems when the

M. Elad and M. Aharon. Image denoising via sparse and redundant representations over


